Delator: Detecção Automática de Indícios de Lavagem de Dinheiro por Redes Neurais em Grafos de Transações

Henrique S. Assumpção[1], **Fabrício Souza**[1], **Leandro Lacerda Campos**[1,2], **Vinícius T. de Castro Pires**[1,2], **Paulo M. Laurentys de Almeida**[2], **Fabricio Murai**[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais

[2]InterMind – Inter S.A.

{henriquesoares, fabricio.souza, murai}@dcc.ufmg.br

{leandro.campos, vinicius.pires, paulo.laurentys}@bancointer.com.br

Abstract. *Money laundering is one of the most relevant criminal activities today, due to its potential to cause massive financial losses to governments, banks, etc. We propose DELATOR, a new CAAT (computer-assisted audit technology) to detect money laundering activities based on neural network models that encode bank transfers as a large-scale temporal graph. In collaboration with a Brazilian bank, we design and apply an evaluation strategy to quantify DELATOR's performance on historic data comprising millions of clients. DELATOR outperforms an off-the-shelf solution from Amazon AWS by 18.9% with respect to AUC. We conducted real experiments that led to discovery of 8 new suspicious among 100 analyzed cases, which would have been reported to the authorities under the current criteria.*

Resumo. *A lavagem de dinheiro é hoje uma das atividades criminais mais relevantes, principalmente devido ao seu potencial de provocar grandes perdas financeiras para governos, bancos, etc. Nós propomos o DELATOR, uma nova CAAT (tecnologia de auditoria assistida por computador) para detectar atividades de lavagem de dinheiro baseada em modelos de rede neural que codificam transferências bancárias como um grafo temporal de larga escala. Em colaboração com um banco brasileiro, projetamos e aplicamos uma estratégia de avaliação para quantificar o desempenho do DELATOR em dados históricos que englobam milhões de clientes. O DELATOR supera uma solução off-the-shelf da Amazon AWS em 18.9% em termos de AUC. Conduzimos experimentos reais que levaram à descoberta de 8 novos casos suspeitos dentre 100 analisados, que seriam reportados às autoridades sob os critérios atuais.*

1. Introdução

Lavagem de dinheiro é um termo utilizado para se referir a processos que tentam legitimar dinheiro obtido de formas ilegais, como, por exemplo, tráfico de drogas, sonegação de impostos, furtos, etc [FATF 2021]. Tipicamente, a lavagem de dinheiro ocorre através de uma sucessão de transações financeiras que ofuscam a origem do dinheiro. Em 2017, o Brasil foi considerado líder mundial em lavagem de dinheiro: a prática foi testemunhada em 23% das companhias nacionais, enquanto a média global era 16%.[1] Entre 2013 e 2017, a Polícia Federal contabilizou mais de R$ 123 bilhões de prejuízos ao país.

[1]veja.abril.com.br/economia/brasil-e-o-maior-do-mundo-em-lavagem-de-dinheiro/; blog.idwall.co/o-que-e-pld-prevencao-lavagem-dinheiro/

A maioria dos países ocidentais define um regulamento a ser seguido pelos bancos, que é baseado em cenários que podem caracterizar lavagem de dinheiro e financiamento ao terrorismo [FATF 2021], por exemplo: receber um depósito de valor muitas vezes maior do que a renda declarada. Com base nesses cenários, os bancos implementam procedimentos de monitoramento, seleção, análise e comunicação de operações e situações suspeitas. No Brasil, tais procedimentos são regulamentados pelo Banco Central do Brasil por meio da Circular no 3.978 [BACEN 2020a], de 23 janeiro de 2020, e da Circular no 4.001 [BACEN 2020b], de 29 de janeiro de 2020. Em um cenário de rápida expansão da carteira de clientes e da oferta de serviços bancários digitais, a efetividade dos procedimentos de monitoramento e seleção passa a ser de fundamental importância.

Na prática, esse monitoramento é feito pelos bancos usando sistemas que implementam uma série de regras. Essas regras variam em natureza, mas muitas vezes correspondem a limites numéricos para sinalizar transações suspeitas. As regras geram alertas para casos suspeitos, que então precisam ser manualmente inspecionados por especialistas a fim de decidir quais devem ser encaminhados às autoridades (no Brasil, ao Conselho de Controle de Atividades Financeiras, COAF). Os sistemas de monitoramento atuais sofrem de dois problemas principais. O primeiro é a geração de um volume muito grande de alertas que precisam ser analisados por times dedicados à prevenção à lavagem de dinheiro. O segundo advém do uso de regras estáticas no sistema de monitoramento, que faz com que casos próximos à região de fronteira dos limites numéricos implementados não sejam enviados para análise. Isto não seria problema caso os regulamentos especificassem estes limites, mas eles apresentam apenas uma descrição em alto nível dos cenários que caracterizam os crimes, deixando sua implementação concreta a cargo de cada banco.

A solução para os problemas anteriores requer o desenvolvimento de CAATs (*computer-assisted audit technologies*). Um dos maiores desafios na proposta de novas CAATs é a sua avaliação. Além da dificuldade de se obter acesso a esses dados por questões de privacidade e segurança, avaliar o desempenho de um modelo que propõe novos casos para a análise requer o esforço de especialistas no domínio. Embora existam alguns poucos datasets públicos para detecção de fraudes envolvendo criptomoedas [Weber et al. 2019], o domínio de movimentação bancária apresenta diferenças importantes, como a grande variedade do tipo de transações.

Neste contexto, propomos o DELATOR, um novo framework computacional para detecção de lavagem de dinheiro e de financiamento ao terrorismo, com o objetivo de resolver os problemas descritos anteriormente. O DELATOR é baseado no grafo de transações bancárias. Este trabalho foi realizado em colaboração com o InterMind, laboratório de IA do Inter, o que permitiu realizar um estudo de caso utilizando a nova metodologia de avaliação proposta aqui. Neste estudo, o framework proposto foi aplicado a alguns milhões de contas, sugerindo 100 casos para análise, o que levou à descoberta de 8 casos que seriam encaminhados ao COAF segundo as regras atuais, tornando mais assertivo o trabalho analítico já desempenhado atualmente. **Contribuições deste trabalho:**

1. Um arcabouço geral para a detecção de casos de lavagem de dinheiro e financiamento ao terrorismo baseado em transações financeiras;
2. Um novo modelo para extração de representações vetoriais de clientes baseado em aprendizado multi-tarefa (*multi-task learning*) sobre grafos dinâmicos; e
3. Uma metodologia híbrida de avaliação para esta tarefa; parte offline e parte online.

2. Trabalhos Relacionados

Nesta seção revisamos os principais trabalhos em CAATs, detecção de fraude em transações financeiras e, em particular, detecção de fraudes usando modelos em grafo.

Técnicas de Auditoria Assistidas por Computador. CAATs (Computer-assisted audit technologies) têm sido propostas para tornar mais eficaz o processo de auditoria em várias áreas, incluindo o de fraudes financeiras [Widuri and Gautama 2020, Othman et al. 2015]. Tais tecnologias diferem da detecção automática porque pressupõem a atuação humana para a tomada de decisão. Isto é importante em contextos onde falsos positivos têm consequências muito negativas e onde frequentemente necessita-se a obtenção de informações extras para a tomada de decisão. As CAATs são baseadas em modelos para detecção de fraude e inconsistências, tais como os discutidos a seguir.

Modelos para detecção de fraude em transações financeiras. Em 1998 foi patenteado o primeiro modelo de sistema para detecção automática de transações fraudulentas em meios de pagamento eletrônicos [Gopinathan et al. 1998]. No modelo, as características que discriminam as transações ilícitas são extraídas a partir do relacionamento entre variáveis relevantes ao domínio. Hoje em dia, CAATs são baseados em mineração de dados, análise digital, ou software de verificação de fraudes, e têm papel importante na detecção de fraude [Halbouni et al. 2016]. Em particular, a detecção de lavagem de dinheiro pode se beneficiar da representação da rede de transações como grafos.

Modelos para detecção de fraudes com modelos baseados em grafo. Avanços recentes na área de aprendizado de representações em grafos impulsionados pelas Redes Neurais em Grafos (GNNs, do inglês Graph Neural Networks) têm levado ao surgimento de novos estados-da-arte em tarefas de predição em redes, tais como classificação de nós, predição de links, previsão de propriedades dos grafos, etc. Estas representações vetoriais, conhecidas como *embeddings*, também vêm sendo cada vez mais exploradas para a detecção de atividades suspeitas [Wang et al. 2021]. Neste cenário, os dois desafios principais são o desbalanceamento extremo de dados e o desvio de conceito (i.e., aquilo que caracteriza uma "anomalia" pode mudar ao longo do tempo). As GNNs costumam sofrer do efeito conhecido como *oversmoothing* quando as classes são desbalanceadas, isto é, os embeddings das diferentes classes tendem a ficar muito similares [Zhao et al. 2021].

Algumas arquiteturas GNN propostas modificam as funções de perda, supervisionada [Liang et al. 2019] ou semissupervisionada [Wang et al. 2019], para lidar com o desbalanceamento de classe. Nesta mesma direção, os autores de [Weber et al. 2019] propuseram novas arquiteturas Skip-GCN e EvolveGCN, capazes de superar o desempenho de uma Graph Convolutional Network (GCN). Em particular, a EvolveGCN [Pareja et al. 2020] tenta lidar com o desvio de conceito usando redes neurais do tipo LSTM (long short-term memory). O Pick-and-Choose é uma técnica recente que tenta incluir arestas virtuais entre nós da classe minoritária [Liu et al. 2021]. No entanto, os autores de [Pereira and Murai 2021] mostram que essa técnica não tem um desempenho tão bom em datasets mais desbalanceados usando datasets reais e sintéticos gerados pelo simulador de transações AMLSim. Outras arquiteturas não tratam do desbalanceamento de classe em tarefas de detecção de fraude, mas são projetadas para redes heterogêneas [Zhu et al. 2020]. Como diferencial deste trabalho, propomos GNNs treinadas a partir de duas tarefas (previsão de links e do valor transferido) para detecção da fraude conhecida como lavagem de dinheiro, utilizando dados reais.

3. Visão geral do framework

O framework proposto neste trabalho é uma CAAT cuja finalidade é auxiliar times de PLD de instituições financeiras na auditoria de casos suspeitos de lavagem de dinheiro e financiamento ao terrorismo. A tarefa de classificação a ser resolvida pelo framework é: *"dadas as informações cadastrais e transacionais de um cliente, qual a probabilidade de ele ser enquadrado em um caso suspeito e por isso ser encaminhado ao COAF?"*. O treinamento de um modelo para esta tarefa consiste em três etapas:

1. Obtenção de atributos brutos: obtidos para cada cliente (e.g., renda, profissão, etc).

2. Extração de representações numéricas (features): tradicionalmente, a extração de features é feita a partir da transformação dos atributos brutos associados a cada observação. Essas transformações podem ser definidas manualmente com uso de conhecimento do domínio de aplicação. Para cada cliente i, obtém-se um vetor numérico $\mathbf{x}_i$. Alternativamente, as features podem ser obtidas através de técnicas de embedding, que têm como objetivo aprender automaticamente o vetor numérico que melhor representa um cliente, a partir de informações de contexto. Neste trabalho, dados sobre transações realizadas por um cliente podem ser usados para definir o seu "contexto". O conjunto de todas as transações será usado para construir um grafo de transações, de diferentes formas. A partir do grafo, usaremos métodos projetados para aprender embeddings de nós. Para cada cliente i, obtém-se um vetor real $\mathbf{z}_i$, que pode ou não ser concatenado às transformações descritas anteriormente para gerar a representação final $\mathbf{x}_i$ do cliente.

3. Treinamento do classificador: uma vez obtidas as features que representam cada cliente, podemos usar dados rotulados para treinar modelos de classificação. Estes modelos aprendem a mapear um vetor de features $\mathbf{x}_i$ à probabilidade de que um cliente receba um determinado rótulo (e.g., y_i = "encaminhado"). Tais probabilidades podem ser usadas juntamente com um *threshold* (limiar) para retornar uma previsão (e.g., se $\Pr(y_i = 1|\mathbf{x}_i) > 0{,}5$ então prevê "encaminhado"). Optamos por utilizar o classificador LightGBM [Ke et al. 2017] por sua escalabilidade e capacidade de obter resultados do estado da arte para diversas tarefas de aprendizado supervisionado.

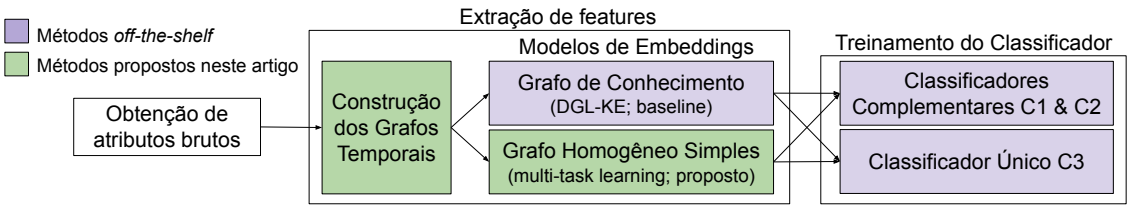


Figura 1. Visão Geral das quatro soluções investigadas.

4. Detalhamento das soluções investigadas

A Figura 1 provê uma visão geral das quatro soluções investigadas neste projeto, à luz das etapas descritas em §3. Na extração de features, primeiro construímos uma sequência de grafos temporais (§4.1). Estes grafos são usados como entrada para um de dois métodos de embeddings em grafos (sendo o 1^o uma solução *off-the-shelf* baseada em grafos de conhecimento, e o 2^o uma solução proposta neste trabalho; §4.2). No treinamento do classificador, consideramos duas abordagens possíveis (§4.3): dois classificadores complementares, ou um classificador único. Considerando as combinações possíveis dos métodos de embeddings e abordagens de classificação, iremos avaliar quatro soluções para esta tarefa.

4.1. Construção dos grafos temporais

Modelos clássicos consideram uma visão "individual" de cada cliente, i.e., dados cadastrais dos clientes, tipos de transações, valores, etc. No entanto, considerar a rede definida pelas transações financeiras entre indivíduos pode permitir a obtenção de features mais informativas para cada cliente. Esta rede é representada pela abstração matemática conhecida como grafo. Neste grafo, os indivíduos são representados por nodos/nós (ou vértices) e as transações, por arestas (ou links) que conectam os nós. A vantagem dessa representação é permitir uma visão mais completa dos dados: no exemplo ilustrado pela Figura 2, uma transferência de valor alto de Ana (nó A) para Bruno (nó B) pode não ser suspeita se Bruno costuma receber depósitos altos, enquanto uma transferência de Ana para Carlos (nó C) no mesmo valor pode ser de interesse para análise.

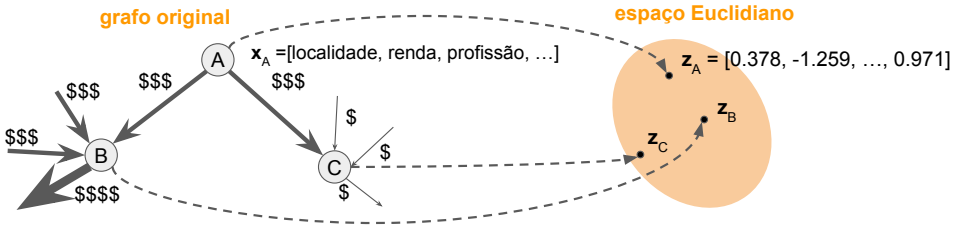


Figura 2. Redes Neurais em Grafos permitem a obtenção de *embeddings* (representações vetoriais) dos nós que compõem um grafo.

Transações financeiras têm duas características importantes: podem ser de muitos tipos diferentes e estão associadas a um dia útil específico. Portanto, os grafos resultantes são heterogêneos e dinâmicos. Devido ao grande número de tipos de aresta, pode-se dizer que são grafos de conhecimento. Neste trabalho utilizamos dados reais de clientes (anonimizados) obtidos junto ao Inter. Não iremos apresentar uma caracterização detalhada dos dados por questões de confidencialidade.

O modelo em grafos mais geral que utilizamos é um modelo heterogêneo e dinâmico, onde as transações são agrupadas por semana. Cada semana é um *snapshot* do grafo. Este modelo é ilustrado na Figura 3. Tal agrupamento é vantajoso em relação à alternativa mais fina (i.e., agrupamento em dias), pois não é sensível às grandes variações observadas entre os dias de semana (e.g., compare a segunda-feira, em que são registradas transações que ocorreram também durante o final de semana, com uma terça-feira). **Iremos construir embeddings que representam um mesmo nó para cada semana e concatená-los, a fim de obter sua representação temporal.**

4.2. Modelos de embedding para grafos

Utilizamos *Graph Neural Networks* (GNNs) para a extração automática de features dos clientes, que levam em consideração a estrutura do grafo, podendo também considerar atributos dos nós e das arestas. Consideramos dois modelos para essa extração:

Modelo de grafo de conhecimento (baseline): a primeira abordagem, utilizada como baseline, é baseada em uma solução *off-the-shelf* implementada pela biblioteca DGL-KE (Deep Graph Library - Knowledge Embedding) proposta por um dos times da Amazon AWS [Zheng et al. 2020]. Embora o modelo seja relativamente sofisticado por mapear cada tipo de transação para um tipo de aresta diferente e por considerar valores nas arestas, apresenta duas desvantagens principais: (i) os valores das transações são interpretados

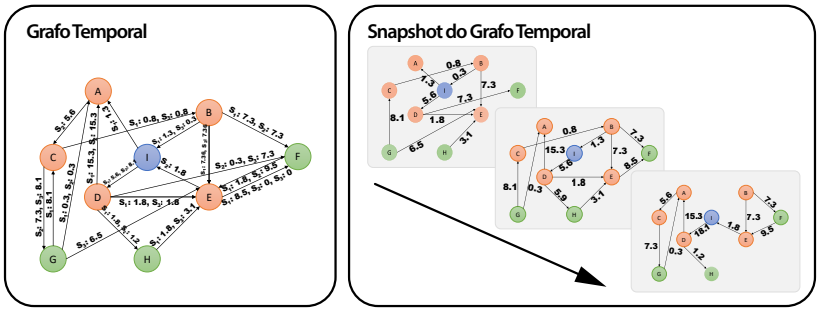


Figura 3. Criação dos snapshots do grafo temporal a partir da agregação das transações que ocorreram na mesma semana.

como "pesos" das arestas e, portanto, clientes que transacionam valores altos entre si tendem a gerar representações mais similares, o que nem sempre é desejável; (ii) o modelo não leva em consideração os atributos dos nós. Para contornar o problema (ii), concatenamos os atributos dos nós às representações vetoriais retornadas pela DGL-KE. As representações finais seguem o formato $\mathbf{x}_i = [\mathbf{z}_i \| \text{localidade}, \text{renda}, \text{profissão}, \ldots]$, onde $\mathbf{z}_i$ é a representação vetorial do nó i gerada pela DGL-KE.

Modelo de grafo homogêneo simples com multi-task learning (proposto): a segunda abordagem, proposta aqui, leva em conta as particularidades da aplicação. Apesar de usar uma versão simplificada dos dados por agregar todas as arestas direcionadas entre um mesmo par de vértices pela soma de seus valores (i.e., todas as transações de i para j são somadas, independentemente de seu tipo), ela confere a semântica correta aos valores associados às arestas ao definir uma tarefa de aprendizado secundária descrita a seguir.

Em geral, embeddings de nós são aprendidos de maneira não-supervisionada com base na tarefa de predição (da presença ou ausência) de links entre nós. Neste projeto, utilizamos a técnica GraphSAGE [Hamilton et al. 2017] para obter essa representação para cada nó i, que será denotada por $\mathbf{h}_i^{(uns)}$ para indicar *unsupervised learning*. Além disso, definimos uma segunda etapa de aprendizado (desta vez supervisionada) com base na tarefa de regressão do valor associado às arestas existentes. Novamente, utilizamos o GraphSAGE para obter uma segunda representação para cada nó i, denotada por $\mathbf{h}_i^{(reg)}$, que será usada junto a $\mathbf{h}_i^{(uns)}$. Para fazer a predição do valor transacionado de i para j, utilizamos um multi-layer perceptron, que recebe como entrada os vetores $\mathbf{x}_i = [\mathbf{h}_i^{(uns)} \| \mathbf{h}_i^{(reg)}]$ e $\mathbf{x}_j = [\mathbf{h}_j^{(uns)} \| \mathbf{h}_j^{(reg)}]$, onde $\mathbf{h}_{(\cdot)}^{(uns)}$ é um embedding fixo, aprendido na etapa não-supervisionada, e $\mathbf{h}_{(\cdot)}^{(reg)}$ é um embedding que varia conforme os parâmetros da GNN treinada durante a etapa supervisionada (regressão), e $\|$ representa concatenação.

A combinação de múltiplas tarefas de aprendizado é conhecida na literatura como **multi-task learning** [Radford et al. 2018]. Os embeddings de nós obtidos dessa forma capturam tanto a probabilidade de que uma conta transacione com a outra, quanto a magnitude dos valores prováveis de serem transferidos.

4.3. Treinamento do Classificador

Abordagem com dois classificadores complementares Inicialmente, definimos um arcabouço de predição que utiliza 2 classificadores complementares, fortemente inspirado pelo mecanismo existente de análise de casos suspeitos. O primeiro classificador (C1) tenta prever se um indivíduo será considerado suspeito ou não, com base nos indícios, en-

quanto o segundo (C2) tenta prever se um caso já considerado suspeito será encaminhado ao COAF ou não, utilizando apenas os indivíduos indicados pelo classificador C1.

Abordagem com classificador único O uso de dois modelos separados torna o problema mais simples e permite avaliar individualmente o desempenho em cada etapa. Isso também reduz drasticamente o problema de desbalanceamento de dados. Contudo, investigamos também o uso de um único classificador (C3) que tem por objetivo calcular diretamente a probabilidade de que um caso seja encaminhado ao COAF.

Em ambos os casos, tendo em vista a baixa quantidade de indivíduos considerados suspeitos, utilizamos a técnica SMOTE [Fernández et al. 2018], que gera observações sintéticas da classe minoritária por meio de interpolação. Em todos os casos, utilizamos o CatBoost como classificador de estado-da-arte para dados com muitos atributos categóricos [Prokhorenkova et al. 2018], treinado a partir de splits do conjuntos de dados descritos em detalhe no relatório técnico [Assumpção et al. 2022].

5. Metodologia de Avaliação

Nesta seção descrevemos a metodologia usada para avaliar as quatro soluções apresentadas na Seção 4. A avaliação é dividida em duas partes, uma *offline* e outra *online*. A avaliação *offline* recebe esse nome porque é realizada usando apenas dados históricos, i.e. de análises passadas. A segunda é dita *online* porque necessita que o time de PLD analise novos casos sugeridos pelas soluções (i.e., casos que não foram enquadrados nas regras estáticas e que, portanto, não haviam sido avaliados).

5.1. Avaliação Offline

Com base em dados históricos, a avaliação offline visa responder às seguintes questões:

- Q1: C1 consegue distinguir casos suspeitos e não-suspeitos?
- Q2: C2 consegue distinguir, dentre os casos suspeitos, aqueles que não serão encaminhados ao COAF (auto close)?
- Q3: C3 consegue distinguir, dentre todos os clientes, aqueles que não serão encaminhados ao COAF (auto close)?

Para responder Q1, usaremos o classificador C1 para prever a probabilidade de que cada cliente seja considerado suspeito, com base nas movimentações de dois meses consecutivos de um ano anterior. Para Q2, usaremos C2 para prever, dentre os casos analisados, qual a probabilidade de que eles sejam encaminhados ao COAF. Deseja-se que a probabilidade associada aos casos não-encaminhados seja muito baixa. Para Q3, usaremos C3 para prever, dentre todos os clientes, qual a probabilidade de que eles sejam encaminhados ao COAF. Novamente, deseja-se que a probabilidade associada aos casos indiciados mas não-encaminhados seja muito baixa. As questões Q2 e Q3 se referem à funcionalidade chamada **auto close**, que permitiria fechar casos pouco prováveis automaticamente.

Métricas de Desempenho A matriz de confusão é um instrumento bastante útil para caracterizar o desempenho de um método de classificação. Além de usarmos a matriz de confusão para ilustrar os resultados, vamos utilizar as seguintes métricas unidimensionais de desempenho para comparar os métodos:

- **F1-fraud:** média geométrica entre precisão e revocação, para "fraude" = rótulo 1.
- **Macro F1:** média simples entre o F1-fraud (classe 1) e o F1 da classe 0.

- **F1-fraud (max):** valor máx. de F1-fraud considerando todos thresholds possíveis.
- **AUC:** área abaixo da curva ROC (taxa verdadeiro positivo vs. taxa falso positivo).
- **AUPR:** área abaixo da curva precisão-revocação. É mais informativa que a AUC nos casos em que há desbalanceamento de classe, como no nosso caso.

Configurações experimentais Os dados usados nos experimentos com os classificadores C1 e C3 foram os embeddings de clientes Inter do tipo pessoa física, separados entre treino/validação e teste. Para os experimentos com C2, consideramos apenas os clientes rotulados como suspeitos pelas regras de PLD. Detalhes sobre as configurações dos experimentos e modelos podem ser encontrados em nosso relatório técnico [Assumpção et al. 2022].

5.2. Avaliação Online

A avaliação online será feita com o auxílio do time de PLD pois necessita gerar rótulos para casos que não foram considerados suspeitos segundo as regras estáticas e, portanto, não foram analisados anteriormente. Esta análise visa responder às seguintes questões:

- Q4: C1 e C2 podem ser usados, em conjunto, para sugerir casos interessantes a serem analisados dentre aqueles que não são considerados suspeitos segundo as regras atualmente implementadas (auto open)?
- Q5: C3 pode ser usado para sugerir casos interessantes a serem analisados dentre aqueles que não são considerados suspeitos segundo as regras atualmente implementadas (auto open)?

Para a questão Q4, iremos considerar casos muito prováveis de serem considerados suspeitos segundo C1 e, simultaneamente, muito prováveis de serem encaminhados ao COAF segundo C2, mas que não caíram em nenhuma regra. Selecionamos 50 casos a serem avaliados pelo time de PLD sugeridos a partir da combinação "modelo baseline + dois classificadores". Para Q5, iremos considerar casos muito prováveis de serem encaminhados ao COAF segundo C3, mas que não caíram em nenhuma regra. Selecionamos 50 casos a serem avaliados pelo time de PLD sugeridos a partir da combinação "modelo proposto + classificador único". Estas questões se referem à funcionalidade chamada **auto open**, que permitiria propor para análise casos que não caíram nas regras estáticas.

Métricas de desempenho Número de casos da lista que seriam encaminhados ao COAF.

6. Resultados

Os resultados serão apresentados em porcentagem por razões de confidencialidade.

6.1. Avaliação Offline

Na avaliação offline, fizemos experimentos com as quatro combinações possíveis envolvendo (i) os modelos de grafo (baseline DGL-KE vs. modelo proposto de multi-task learning) e (ii) as abordagens de classificação (classificadores complementares C1+C2 vs. classificador único C3). Descrevemos abaixo os resultados obtidos.

Q1: C1 consegue distinguir casos suspeitos e não-suspeitos? A Tabela 1 mostra as matrizes de confusão obtidas para o baseline (esquerda) e para o modelo proposto (direita), para o threshold $p > 0{,}5$. Observamos que o baseline gera muitos falsos positivos. Os

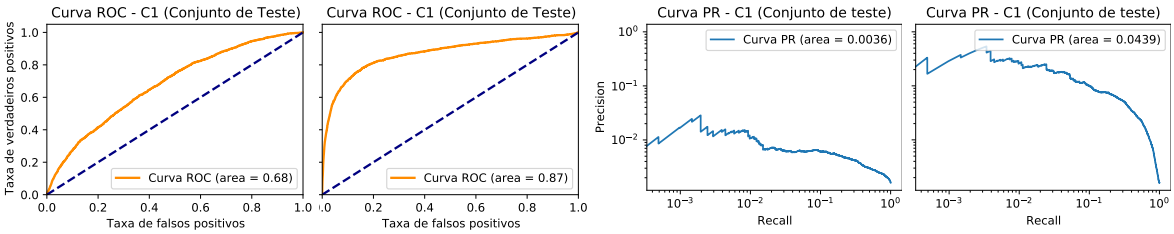
baseline		previsto não susp.	previsto suspeito
real	não susp.	72,179%	27,661%
	suspeito	0,078%	0,082%

proposto		previsto não susp.	previsto suspeito
real	não susp.	94,795%	5,044%
	suspeito	0,065%	0,095%

Tabela 1. Matrizes de Confusão para Q1.

	F1-fraud	F1-fraud (max)	Macro F1	AUC	AUPR
baseline	0,006	0,012	0,422	0,680	0,0036
proposto	0,036	0,108	0,505	0,869	0,0439

Tabela 2. Métricas de desempenho para Q1 (quanto maior, melhor).



(a) baseline **(b)** proposto **(c)** baseline **(d)** proposto

Figura 4. Curvas ROC e Precisão-Revocação de cada modelo, para Q1.

falsos positivos indicam casos que não caem em nenhuma regra, mas que foram previstos pelo modelo como suspeitos. Esses casos podem ser propostos para análise, desde que não correspondam a um volume muito grande. Nesse contexto, a proporção de casos propostos pelo baseline é alta (27,7%), quando comparada com a do modelo proposto (5,1%). Os falsos negativos, por sua vez, não são um problema, pois seriam encontrados pelas regras estáticas de PLD.

Já a Tabela 2 lista as métricas obtidas por cada modelo. Observamos que o modelo proposto supera o baseline por uma margem grande. Com um AUC 0,869, o modelo proposto não só consegue identificar bem os casos que caem em algum indício, mas também pode ser útil para encontrar aqueles que estão próximos às condições limítrofes implementadas pelas regras. O AUPR, que é um bom indicador de desempenho da classificação em dados desbalanceados, é uma ordem de magnitude maior para o modelo proposto.

As Figuras 4(a-b) e (c-d) ilustram, respectivamente, as curvas ROC e Precisão-Revocação de cada modelo, atestando a superioridade do modelo proposto.

baseline		previsto não enc.	previsto encamin.
real	não enc.	91,606%	7,077%
	encamin.	1,318%	0,000%

proposto		previsto não enc.	previsto encamin.
real	não enc.	74,622%	24,061%
	encamin.	0,976%	0,342%

Tabela 3. Matrizes de Confusão para Q2.

Q2: C2 consegue distinguir, dentre os casos suspeitos, aqueles que não serão encaminhados ao COAF (auto close)? A Tabela 3 mostra as matrizes de confusão obtidas

	F1-fraud	F1-fraud (max)	Macro F1	AUC	AUPR
baseline	0,000	0,028	0,478	0,401	0,0099
proposto	0,026	0,029	0,441	0,491	0,0115

Tabela 4. Métricas de desempenho para Q2 (quanto maior, melhor).

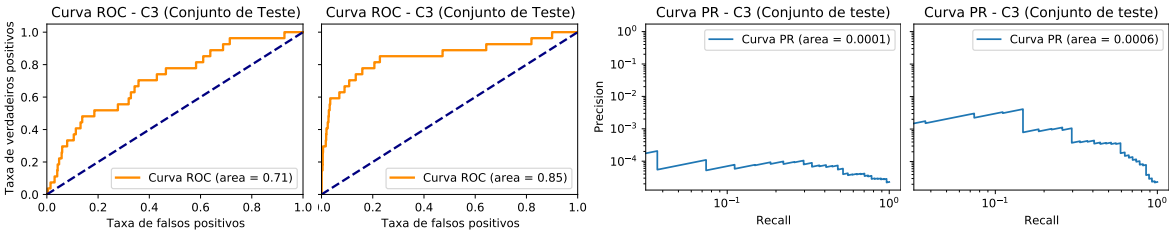
para o baseline (esquerda) e para o modelo proposto (direita). Para ambos os modelos, a taxa de falsos negativos é bastante alta. Os resultados mostrados na Tabela 4, que lista as métricas obtidas por cada modelo, são consistentes com esta observação. Caso o AUC dos modelos fosse mais alto, poderia haver um threshold de classificação abaixo do qual os casos seriam muito pouco prováveis de serem encaminhados. Infelizmente este não é o caso. Analisamos também as curvas ROC e Precisão-Revocação (omitidas por limitações de espaço), e observamos que não exibem as características desejadas.

baseline		previsto não enc.	previsto encamin.	**proposto**		previsto não enc.	previsto encamin.
real	não enc.	97,977%	2,021%	real	não enc.	99,815%	0,183%
	encamin.	$20 \cdot 10^{-4}$%	$1{,}6 \cdot 10^{-4}$%		encamin.	$18 \cdot 10^{-4}$%	$3{,}1 \cdot 10^{-4}$%

Tabela 5. Matrizes de Confusão para Q3.

	F1-fraud	F1-fraud (max)	Macro F1	AUC	AUPR
baseline	0,0001	0,0004	0,4941	0,714	0,0001
proposto	0,003	0,008	0,501	0,849	0,0006

Tabela 6. Métricas de desempenho para Q3 (quanto maior, melhor).



(a) baseline **(b)** proposto **(c)** baseline **(d)** proposto

Figura 5. Curva ROC e Precisão-Revocação de cada modelo, para Q3.

Q3: C3 consegue distinguir, dentre todos os clientes, aqueles que não serão encaminhados ao COAF (auto close)? A Tabela 5 mostra as matrizes de confusão obtidas para o baseline (esquerda) e para o modelo proposto (direita). Nota-se que o modelo proposto identifica o dobro de casos encaminhados, e ainda exibe bem menos falsos positivos (2% vs. 0,2%). A Tabela 6 lista as métricas obtidas por cada modelo. O desempenho relativo dos modelos neste experimento é similar àquele observado no experimento associado a Q1: observamos que o modelo proposto supera o baseline por uma margem grande. Com um AUC de 0,849, o modelo proposto se mostra bem promissor na identificação de casos a serem encaminhados. Ademais, o AUPR é 6 vezes maior para o modelo proposto.

As Figuras 5(a-b) e (c-d) ilustram, respectivamente, as curvas ROC e Precisão-Revocação de cada modelo. Em geral, observamos que a curva não assume valores de precisão muito altos. A queda abrupta observada no lado extremo esquerdo da Figura 5(d) indica que é promissor analisar os casos previstos como mais prováveis pelo classificador C3 treinado com os embeddings do modelo proposto.

6.2. Avaliação Online

Vimos que na avaliação offline o melhor resultado foi obtido pela combinação do modelo proposto com C3. O mesmo resultado foi observado na avaliação online, realizada com o auxílio do time de PLD: (Q4) o baseline combinado com C1+C2 nos levou a encontrar **1** novo caso a ser reportado, enquanto (Q5) o modelo proposto com C3 nos levou a encontrar **7** novos casos. Por limitações de espaço, iremos descrever apenas os resultados para Q5.

Q5: C3 pode ser usado para sugerir casos interessantes a serem analisados dentre aqueles que não são considerados suspeitos segundo as regras atualmente implementadas (auto open)? A partir da avaliação feita pelo time de PLD dos 50 clientes enviados para análise, o time concluiu que 7 teriam sido enviados para o COAF.

7. Conclusões e Trabalhos Futuros

Neste trabalho, estudamos o problema de detectar lavagem de dinheiro e financiamento ao terrorismo baseado na rede de transações financeiras entre clientes de um banco, utilizando dados reais do Inter. Introduzimos o framework DELATOR, que consiste em três etapas principais: (i) obtenção de dados brutos, (ii) extração de features e (iii) treinamento do classificador. No contexto de (ii), propusemos uma forma de se construir os grafos e um modelo de aprendizado em grafos homogêneos simples baseado no paradigma multi-task learning. Comparamos o modelo proposto com um baseline *off-the-shelf* conhecido como DGL-KE. O modelo proposto combinado com a abordagem de classificador único obteve os melhores resultados, tanto na avaliação offline quanto na online.[2] Os resultados mostram que a utilização de um CAAT baseado em modelos de aprendizado sobre grafos pode ajudar instituições financeiras ao flexibilizar as regras usadas para detectar indícios de lavagem de dinheiro e financiamento ao terrorismo, e ao propor casos que não foram enquadrados nas regras mas que podem ser interessantes para a análise. O framework pode ser aplicado a outros bancos e a outras aplicações que envolvam grafos de transações financeiras. Como trabalho futuro, iremos testar o DELATOR em outros datasets, incluindo em dados sintéticos gerados pelo AMLSim. O desenvolvimento continuado destes modelos pode levar a grandes progressos no combate a esse tipo de crime.

Referências

Assumpção, H. S., Souza, F., Lacerda Campos, L., Castro Pires, V. T., Laurentys de Almeida, P. M., and Murai, F. (2022). Delator: Automatic detection of money laundering evidence on transaction graphs via neural networks. *arXiv preprint arXiv:2205.10293*.

BACEN (2020a). Circular nº 3.978, de 23 de janeiro de 2020.

BACEN (2020b). Circular nº 4.001, de 29 de janeiro de 2020.

FATF (2012-2021). International standards on combating money laundering and the financing of terrorism & proliferation.

[2]O presente trabalho foi realizado com apoio da FAPEMIG e do time de PLD do Inter.

Fernández, A., Garcia, S., Herrera, F., and Chawla, N. V. (2018). Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *JAIR*, 61:863–905.

Gopinathan, K. M., Biafore, L. S., Ferguson, W. M., Lazarus, M. A., Pathria, A. K., and Jost, A. (1998). Fraud detection using predictive modeling. US Patent 5,819,226.

Halbouni, S. S., Obeid, N., and Garbou, A. (2016). Corporate governance and information technology in fraud prevention and detection. *Managerial Auditing Journal*.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In *NeurIPS*, volume 30.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *NeurIPS*, volume 30.

Liang, C., Liu, Z., Liu, B., Zhou, J., Li, X., Yang, S., and Qi, Y. (2019). Uncovering insurance fraud conspiracy with network learning. In *SIGIR*, pages 1181–1184.

Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., and He, Q. (2021). Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In *WWW*, pages 3168–3177.

Othman, R., Aris, N. A., Mardziyah, A., Zainan, N., and Amin, N. M. (2015). Fraud detection and prevention methods in the malaysian public sector: Accountants' and internal auditors' perceptions. *Procedia Economics and Finance*, 28:59–67.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., and Leiserson, C. E. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *AAAI*.

Pereira, R. and Murai, F. (2021). Quão efetivas são redes neurais baseadas em grafos na detecção de fraude para dados em rede? In *BraSNAM*, pages 205–210.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. In *NeurIPS*, page 6639–6649.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Wang, D., Lin, J., Cui, P., Jia, Q., Wang, Z., Fang, Y., Yu, Q., Zhou, J., Yang, S., and Qi, Y. (2019). A semi-supervised graph attentive network for financial fraud detection. In *ICDM*, pages 598–607.

Wang, J., Zhang, S., Xiao, Y., and Song, R. (2021). A review on graph neural network methods in financial applications.

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., and Leiserson, C. E. (2019). Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*.

Widuri, R. and Gautama, Y. (2020). Computer-assisted audit techniques (caats) for financial fraud detection: A qualitative approach. In *ICIMTech*, pages 771–776.

Zhao, T., Zhang, X., and Wang, S. (2021). Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *WSDM*, pages 833–841.

Zheng, D., Song, X., Ma, C., Tan, Z., Ye, Z., Dong, J., Xiong, H., Zhang, Z., and Karypis, G. (2020). Dgl-ke: Training knowledge graph embeddings at scale. In *SIGIR*, pages 739–748.

Zhu, Y.-N., Luo, X., Li, Y.-F., Bu, B., Zhou, K., Zhang, W., and Lu, M. (2020). Heterogeneous mini-graph neural network and its application to fraud invitation detection. In *ICDM*, pages 891–899.